

# ПОВЫШЕНИЕ ТОЧНОСТИ ОПРЕДЕЛЕНИЯ ЛИНИИ НА ИЗОБРАЖЕНИИ ДЛЯ ОПТИЧЕСКОГО ИЗМЕРИТЕЛЯ

Баранов О.А., Саган А.В., Калмыков А.А.

Уральский федеральный университет имени первого Президента России Б.Н. Ельцина  
проспект Мира, 19, Екатеринбург, Свердловская обл., 620002, Россия  
тел.: 8 (922) 127-77-01, e-mail: wshadw@bk.ru

*Аннотация* — настоящая статья посвящена разработке и реализации алгоритма определения лазерной линии на изображении с субпиксельной точностью. В основе работы алгоритма лежит нахождение субпиксельных координат точек, относящихся к искомой линии, и последующая их аппроксимация методом наименьших квадратов. Субпиксельная точность достигается путем построчного считывания значений яркости пикселей изображения, построению распределения яркостей по ширине линии, нахождению центра в каждом распределении. Работа алгоритма была протестирована на реальных изображениях лазерной подсветки.

## INCREASE OF ACCURACY OF DETERMINATION OF THE LINE ON THE IMAGE FOR THE OPTICAL MEASURING INSTRUMENT

Baranov O.A.1, Sagan A.V.1, Kalmykov A.A.

Ural Federal University named after the first President of Russia B.N. Yeltsin  
pr. Mira, 19, Yekaterinburg, Sverdlovsk region, 620002, Russian Federation  
ph.: 8-922-127-77-01, e-mail: wshadw@bk.ru

*Abstract* — the present article is devoted to development and implementation of algorithm of determination of the laser line on the image with a subpixel accuracy. Finding of subpixel coordinates of the points relating to the required line and their subsequent approximation by the least-squares method is the cornerstone of operation of algorithm. Subpixel accuracy is reached by line-by-line reading of values of brightness of pixels of the image, to creation of distribution of brightness on the line width, finding of center in each distribution. Operation of algorithm was tested on real images of laser illumination.

### I. Введение

Целью данного исследования является нахождение способа увеличения точности определения вертикальной линии на изображении для оптического измерителя.

Необходимо разработать и реализовать такой алгоритм, который позволил бы находить на изображении прямую и определять ее уравнение с субпиксельной точностью (в координатах, не привязанных к пикселям).

### II. Основная часть

#### Выбор алгоритма

В системе технического зрения важнейшим фактором влияющим на результат является алгоритм распознавания, в данном случае двух лазерных линий.

При выборе алгоритма необходимо учесть несколько требований:

- высокая скорость работы (меньше секунды, для возможности использования в реальном времени);
- высокая точность определения уравнения линии;
- применимость к распознаванию лазерных линий.

Алгоритм основанный на преобразовании Хафа

С целью уменьшения объема вычислений (информация о цвете является абсолютно излишней) изображение необходимо преобразовать в полутоновый вид.

Данная операция выполняется по формуле:

$$L = 0,27R + 0,67G + 0,06B, \quad (1)$$

где  $R$  — яркость красной составляющей цветного изображения;  $G$  — яркость зеленой составляющей

цветного изображения;  $B$  — яркость синей составляющей цветного изображения.

Изображение линейно контрастируется по следующей формуле:

$$y = \frac{x - x_{min}}{x_{max} - x_{min}} (y_{max} - y_{min}) + y_{min}, \quad (2)$$

где  $x$  — яркость текущего изображения ( $x_{min}$  — минимальная яркость текущего изображения,  $x_{max}$  — его максимальная яркость);  $y$  — желаемая яркость ( $y_{min}$  — желаемая минимальная яркость,  $y_{max}$  — желаемая максимальная яркость).

Операция контрастирования нужна для увеличения динамического диапазона изображения.

Одной из самых главных проблем обработки изображений является проблема определения контуров, характеризующих границы изображения. Контурные представляют собой такие области на изображении, в которых хорошо заметен сильный контраст интенсивности: резкий скачок интенсивности при переходе к соседнему пикселю. Обнаружение контуров в изображении значительно сокращает объем данных и отфильтровывает ненужную

информацию, сохраняя важные структурные свойства изображения. Задача выделения контуров состоит именно в построении границ объектов и очертаний однородных областей на изображении [1].

Существует несколько фильтров выделения контуров [2]:

а) Линейные:

- 1) По сторонам света.
- 2) Оператор Лапласа.

б) Нелинейные:

- 1) Фильтр Робертса;
- 2) Фильтр Собелла.

Помимо вышеупомянутых фильтров, существует фильтр *Canny*. Данный алгоритм имеет ряд достоинств:

- Небольшой процент ошибок. Очень важно, чтобы присутствующие на изображении границы не были пропущены, а также не было ложных обнаружений;
- Строгая локализованность точек: расстояние между обнаруженными точками контура и фактическими точками границы должно быть минимальным;
- Одно обнаружение на одну границу. Другими словами, исключение возможности неоднократного обнаружения одних и тех же краев.

Преобразование Хафа позволяет на бинарном изображении находить плоские кривые, заданные параметрически, например: прямые, окружности, эллипсы, и т.д. таким образом, как это описано в [3].

Бинарное изображение — изображение, пик-сели которого принимают всего два значения (0 и 1). Считаем 0 — точками фона, 1 — «точками интереса».

Задача преобразования Хафа состоит в выделении кривых, образованных точками интереса.

Ввиду дискретности машинного представления и входных данных (изображения), требуется поэтапно преобразовать непрерывное фазовое пространство в дискретное:

- Вводим сетку на фазовом пространстве;
- Для каждой ячейки сетки устанавливаем соответствующий ей счетчик;
- Значение счетчика каждой ячейки делаем равным количеству точек интереса, через которые проходит хотя бы одна кривая, параметры которой принадлежат данной ячейке;
- Анализ счетчиков ячеек позволяет найти на изображении кривые, на которых лежит наибольшее количество точек интереса.

Прямую на плоскости можно задать следующим образом (рис. 1):

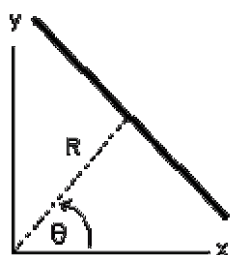


Рис. 1 — Параметрическое отображение прямой

Прямая в этом случае задается следующим уравнением:

$$x \cos(q) + y \sin(q) = R, \quad (2)$$

где  $R$  — длина перпендикуляра опущенного на эту прямую из начала координат;  $q$  — угол между перпендикуляром к этой прямой и осью  $Ox$ ,  $q$  изменяется в пределах от 0 до  $2\pi$ ,  $R$  ограничено размерами входного изображения.

Таким образом, функция, задающая семейство прямых, имеет вид:

$$F(R, q, x, y) = x \cos(q) + y \sin(q) - R, \quad (4)$$

Через каждую точку  $(x, y)$  изображения можно провести несколько прямых с разными  $R$  и  $\theta$ , то есть каждой точке  $(x, y)$  изображения соответствует набор точек в фазовом пространстве  $(R, \theta)$ . В свою очередь каждой точке пространства  $(R, \theta)$  соответствует

набор точек  $(x, y)$  на изображении, образующий прямую рис. 2.

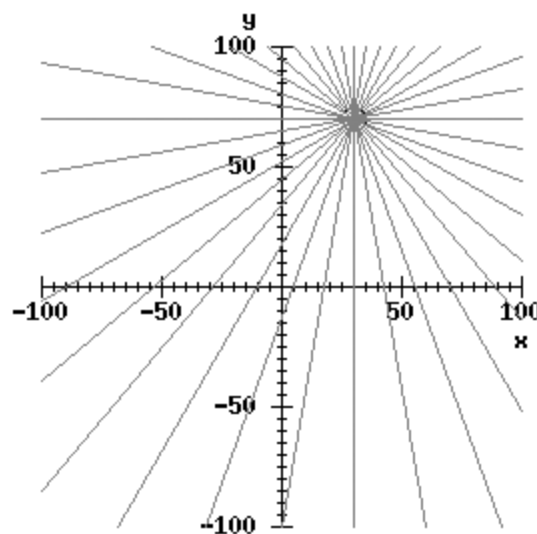


Рис. 2 — Прямые на изображении

Через одну точку можно провести несколько прямых. Учитывая дискретность и введенную сетку, их будет конечное число.

Каждой прямой пространства  $(x, y)$  соответствует точка фазового пространства  $(R, q)$ . Прямые, изображенные на рис. 1, образуют синусоиду рис. 3:

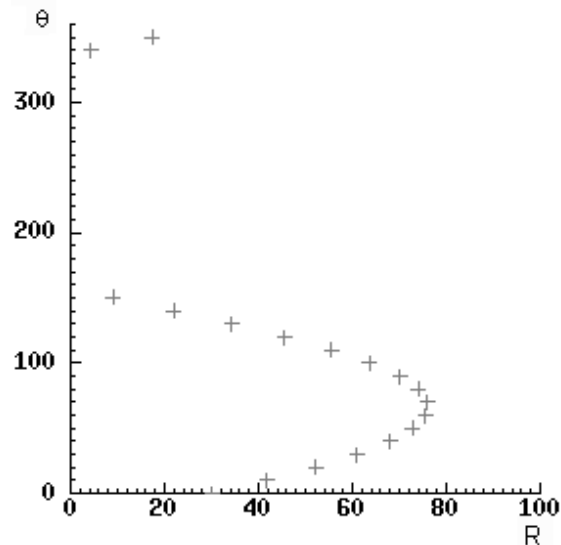


Рис. 3 — Отображение в фазовое пространство

Каждая точка изображения преобразуется в синусоиду. Пусть каждая точка синусоиды имеет значение 1. Тогда, после суммирования и накопления в фазовом пространстве значения данных синусоид, можно построить преобразование Хафа для изображения.

Для того чтобы набор точек стал линией с параметрами  $(R, q)$ , значение в данной точке фазового пространства должно превысить некоторый, задаваемый эмпирически, порог. Но таких точек может быть несколько, и они могут располагаться достаточно близко друг к другу, что

приведет к выделению нескольких прямых с близкими параметрами.

Данный алгоритм распознавания использовался в СТЗ на устройстве измерения зазора валков прошлой конструкции, где использовалось две камеры и светодиодная подсветка.

Уравнение границы валка, которую необходимо было распознать для проведения расчетов, определялось с высокой точностью. В редких случаях были проблемы с определением границы, например при появлении на изображении бликов или засветов.

Основным недостатком данного алгоритма является вычислительная сложность и следовательно низкая скорость работы. При использовании данного алгоритма на прошлой конструкции время работы алгоритма занимало в среднем 10 секунд. Даже после оптимизации путем разделения расчетов на несколько потоков и уменьшения разрешения угла поворота линии удалось снизить время работы алгоритма до 2 секунд, что является недопустимым для использования в реальном времени.

Вторым недостатком является то, что необходимо производить бинаризацию изображения с использованием какого то порога, следовательно теряется часть полезной информации.

При использовании лазерной линии, так как она имеет определенную толщину, то на этапе контуризации будет найдено два контура слева и справа от линии. Следовательно с помощью преобразования Хафа нужно будет искать две линии, а так как лазерных линий две, то и искать нужно будет 4 линии, что сильно скажется на времени работы алгоритма.

Задача распознавания сама по себе является достаточно сложной в плане количества вычислительных операций, поэтому при использовании универсальных решений достичь хороших результатов относительно точности и скорости работы практически не возможно.

Было решено разработать собственный алгоритм распознавания лазерных линий, с максимальной оптимизацией под конкретную задачу использования на измерительном устройстве и который бы максимально соответствовал всем необходимым требованиям.

### Разработка собственного алгоритма

Для точного нахождения вертикальных линий был разработан специальный алгоритм, который позволяет определить положение лазерной линии с субпиксельной точностью.

На первом шаге происходит построчное сжатие изображения путем группировки пикселей. Производится это для устранения большинства шумов и значительного увеличения скорости.

На втором шаге происходит построчное считывание яркости пикселей. При считывании яркости игнорируется синий канал, так как экспериментальным путем было выявлено, что большинство шумов у используемой камеры синего цвета. Далее

производится фильтрация точек с яркостью, попадающей в небольшую область возле минимальной яркости.

Так как анализируется лазерная линия, то значения яркостей тем ниже, чем дальше от центра линии. На этом шаге мы получаем распределения яркостей для каждой строки (рис. 4).

Далее распределения обрезаются в области максимума, размер которой чуть больше ширины лазера, и интерполируется сплайновой интерполяцией [4].

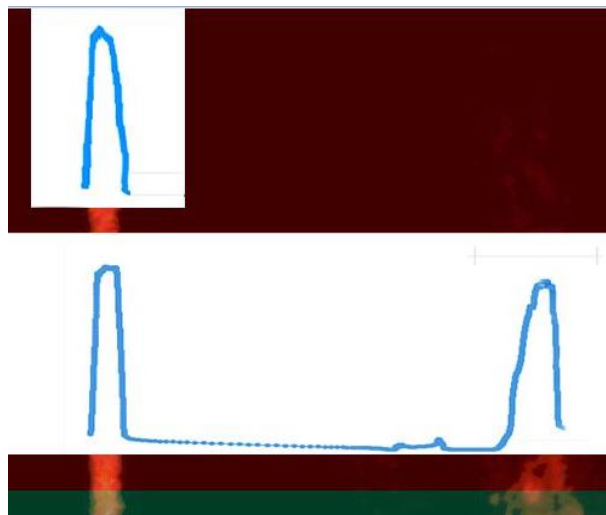


Рис. 4 – Считывание яркостей

$$M[X] = \sum_{i=1}^{\infty} x_i p_i, \quad (1)$$

Эти действия производятся для всех строк, в итоге получается набор точек, по которому строится линия путем аппроксимации методом наименьших квадратов[7] (рис. 5).

Если на изображении будут присутствовать световые пятна, то они могут сместить найденную линию после аппроксимации. Для решения данной проблемы был реализован двухэтапный алгоритм фильтрации таких пятен. Первый этап фильтрации позволяет избавиться от больших засветов на изображении. Второй этап позволяет избавиться от множества небольших, но довольно ярких шумов.

На первом шаге выполняется поиск точек, у которых расстояние до найденной линии больше среднего. Далее определяется, с какой стороны от линии больше точек. Если, например, слева от линии точек больше, то точки справа отбрасываются, а для каждого распределения заново находится ожидаемая координата центра лазера, но уже строго слева от первоначально найденной линии.

На втором этапе фильтрации происходит отсеивание точек с отклонением больше, чем удвоенное среднее. Затем по оставшимся точкам строится линия путём аппроксимации методом наименьших квадратов. (рис. 6).

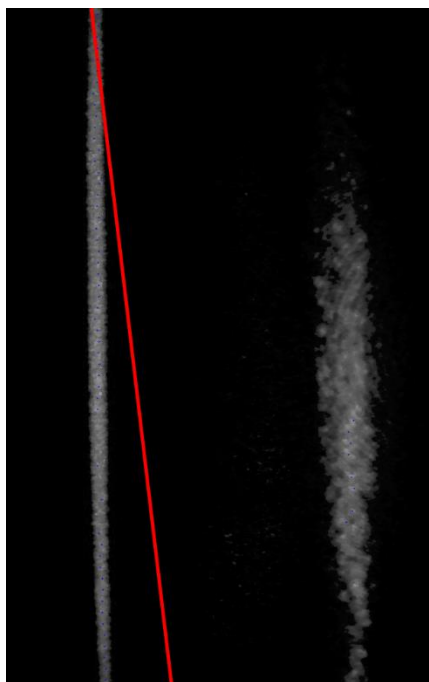


Рис. 5 – Результат аппроксимации

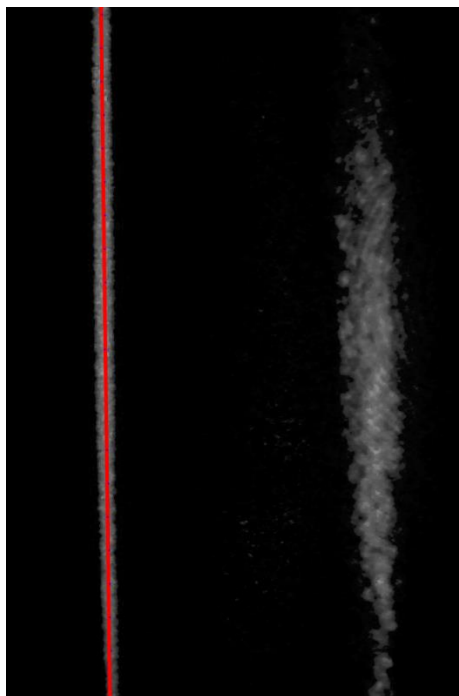


Рис. 6 – Результат фильтрации

Для реализации алгоритма был выбран язык C#. Написана программа, которая позволяет определить линию на входном изображении, вывести ее уравнение, а также посчитать СКО и дисперсию для точек относительно линии. Интерфейс программы представлен на рис. 7.

Для оценки точности определения линии были получены несколько изображений с лазерной линией, расположенной вертикально, чтобы угловой коэффициент  $k$  в уравнении  $y = kx + b$  был приблизительно равен 0 (координатная плоскость на изображении повернута на 90 градусов). Таким образом можно оценивать точность определения линии по координате  $y$ . Результаты представлены в таблице 1.

Таблица 1  
Результаты испытаний

СКО	Дисперсия	Параметр $b$
0,0716	0,00513	275,54
0,1194	0,01424	275,61
0,1196	0,01431	275,60
0,1178	0,01387	275,59
0,0741	0,00552	275,61
0,1094	0,01198	275,53
0,1054	0,01111	275,56

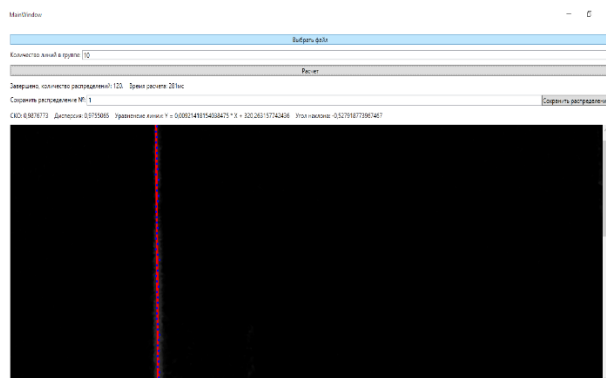


Рис. 7 – Интерфейс программы

### III. Заключение

В ходе работы был разработан алгоритм, который позволяет определять уравнение линии на изображении с субпиксельной точностью.

Из результатов испытаний видно, что точность определения линии выше, как минимум, в десять раз точнее, чем при определении координат в пикселях (разница между самым малым значением параметра  $b$  и самым большим равна 0,08 пикселя).

Работа алгоритма была проверена на реальном высокоточном устройстве. Точность измерения устройства составляет порядка 0,1 мм.

Тот факт, что скорость работы алгоритма высокая и то, что дисперсия параметра  $b$  всего 0,00102, позволяет снимать подряд несколько изображений и усреднять по ним полученные результаты. Это также приведет к увеличению точности, так как позволит снизить влияние редких отклонений.

### IV. Литература

- [1] Билл Грин. Алгоритм выделения контуров Саппу. — Дрексельская лаборатория автоматизированных систем. [Электронный ресурс]. — Электрон. Дан. — Режим доступа: <http://masters.donntu.edu.ua/2010/fknt/chudovskaja/library/article4.htm>
- [2] Прэтт У. Цифровая обработка изображений. Пер. с англ. М.: Мир, 1982. — 312 с.
- [3] Баяковский Ю. М. Курс лекций по компьютерной графике. ВМиК МГУ, 2004.
- [4] Н. Н. Калиткин, Численные методы. М.: Наука, 1978. 44 с.
- [5] В. Феллер, Глава XI. Целочисленные величины. М.: Мир, 1964. 270 с.
- [6] Л. В. Кудрявцев, Краткий курс математического анализа. М.: Наука, 1989. 122 с.
- [7] Ю. В. Линник. Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений. М., 1962. 137 с.